

This document is published in:

Molina, José M. et al. (eds.), 2011. *User-Centric Technologies and Applications*: Proceedings of the CONTEXTS 2011 Workshop (Advances in Intelligent and Soft Computing, 94), Springer, pp. 37-45.
DOI: 10.1007/978-3-642-19908-0_5

© 2011 Springer-Verlag Berlin Heidelberg

Multi-camera control and video transmission architecture for distributed systems

Alvaro Luis Bustamante, José M. Molina, and Miguel A. Patricio

Abstract The increasing number of autonomous systems monitoring and controlling visual sensor networks, make it necessary an homogeneous (device-independent), flexible (accessible from various places), and efficient (real-time) access to all their underlying video devices. This paper describes an architecture for camera control and video transmission in a distributed system like existing in a cooperative multi-agent video surveillance scenario. The proposed system enables the access to a limited-access resource (video sensors) in an easy, transparent and efficient way both for local and remote processes. It is particularly suitable for Pan-Tilt-Zoom (PTZ) cameras in which a remote control is essential.

Key words: multi-camera systems, visual sensor network, video transmission, ptz cameras

1 Introduction

At the moment, the majority of the people still conceives video surveillance systems as synonymous of CCTV systems: people imagine tens of old cameras connected to tens of remote monitors, controlled by tens of bored and unheeding security employers which should pay attention to restricted areas, access doors, people, vehicles, objects and suspicious situations to prevent crimes or disasters. In alternative, many believe that surveillance systems are storage platforms to memorize multimedia data on the environment, video, photos, wiretapped speech, available for human forensic experts to support investigations. This is partially true, and the value of these systems is undoubted.

Applied Artificial Intelligence Group, Universidad Carlos III de Madrid,
Avd. de la Universidad Carlos III, 22, 28270, Colmenarejo, Madrid, Spain
{alvaro.luis,miguelangel.patricio,josemanuel.molina}@uc3m.es
<http://www.giaa.inf.uc3m.es>

However, latest advances in hardware technology and state of the art of computer vision and artificial intelligence research can be employed to develop autonomous and distributed monitoring systems. They are possible and necessary since the enormous improvement and affordability of hardware and the availability of distributed computing technologies have encouraged an increasing use of distributed and parallel systems in monitoring applications [4, 5]. So the growing amount of sensors sometimes makes unaffordable a human monitoring.

In addition, sensors are increasing their uses and capabilities with characteristics like on-board processing, Pan-Tilt-Zoom control, thermal and infrared vision, and so on [14]. It is valuable since it provides more features to the user, but it adds an extra control problem. For instance, the control of PTZ cameras can be easy to achieve in a small scale, but it can become a really tedious task when you have a large vision sensor network, which may lead in a poor usage of the available resources. A sample scenario may be one security employer attempting to monitor a specific moving target from several PTZ cameras simultaneously. The operator should reorientate the cameras in real-time according to the moving object, which result in a hard task to achieve.

This way, it would be useful that all these tedious operations were able to be managed by an autonomous system. It will help the operator who is working with the visual sensor network to exploit efficiently all the resources. Thereby a operator may set a new goal like *'find someone with a red bag'*, and the autonomous system should help in the different devices coordination, in order to meet the established goal.

Many researches has been focused in solve this and similar issues using multi-agent systems [11, 12], in where each agent is the responsible of control and manage one camera. This distributed solution is a good option for the problem of coordinating multi-camera systems, taking the advantages of scalability and fault-tolerance over centralization.

However, many of the proposed theoretical architectures uses to miss the underlying complex task of controlling the video sensors, like the image acquisition and transmission process [12, 8, 13]. They assume that in some way there is a video flow and a control flow for PTZ cameras which can be used in their architectures. However when dealing with a real implementation of a distributed multi-agent architecture with a real visual sensor network the problems arrives.

In order to support this kind of distributed systems, in this this paper is described the required architecture for manage the video devices present in a visual sensor network. This architecture will deal with the PTZ control and the video transmission for each video sensor connected, both for local and remote processes. This way, any camera will have their control and video accessible from any place. It is suitable for multi-agent systems since they really are distributed process and require remote control of the sensors. This kind of architecture involves many disciplines like video compression and transmission, advanced memory management, frame grabbers controllers, etc, thus, an overview and some test of the system designed is presented.

The rest of the paper is organized as follow. In section 2 we describes the environment in which the presented system may work. In section 3 is described the architecture supporting any distributed system. Section 4 contains some performance test, and finally in section 5 some conclusions are presented.

2 Environment overview

Multimedia surveillance systems are an emerging application field requiring multidisciplinary expertise spanning from Signal and Image Processing to Communications and Computer Vision [4]. However, in this paper we focus on the problem of communication between the cameras and the different systems that control them, since cameras are limited-access resources.

Our initial working scenario consists in a visual sensor network in which each video device provides Pant-Tilt-Zoom (PTZ) control. It is intended to be used with a multi-agent system, so we want to provide an homogeneous access between different entities controlling this network.

In this scenario *a priori* may operate over the cameras two different entities. We define a operator which may be the personnel security monitoring the video streamed by different cameras and controlling their orientation. In a multi-agent architecture also exists the agents that generally perform an autonomous control depending on the restrictions/goals imposed by the operator.

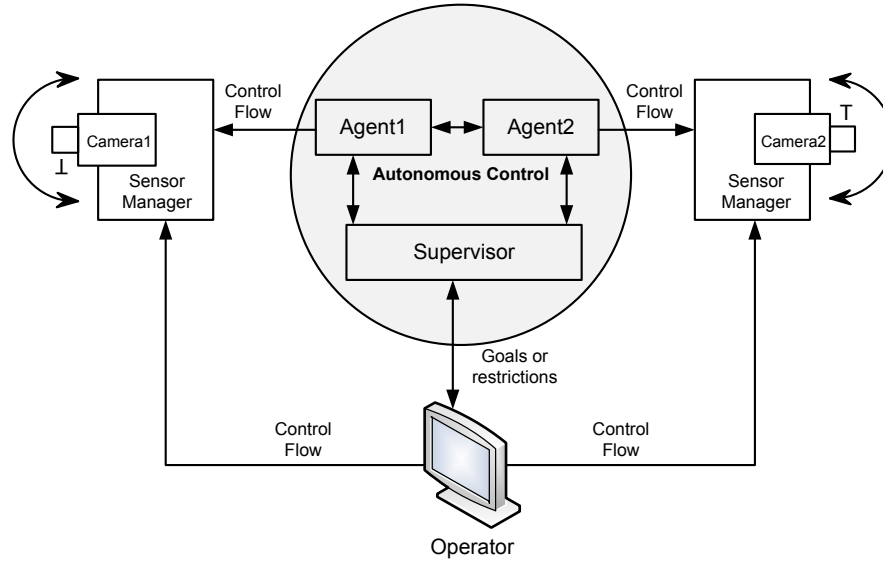


Fig. 1 Example of the environment overview. The camera devices are controlled both by an operator and a distributed system.

This kind of control over the cameras is shown in figure 1, where all the control flow presented in the architecture is described. It is created both from the operator and the multi-agent distributed system, so the devices must be able to handle requests from different sources.

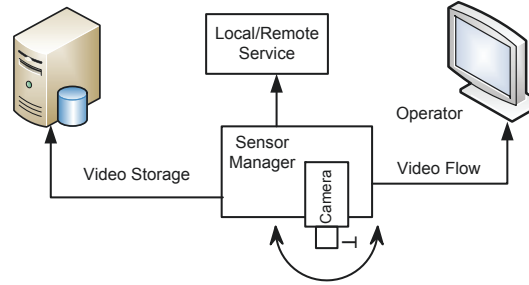


Fig. 2 Camera video transmission to different local and remote systems.

Figure 1 only presents the control flows, but in the architecture also exists video flows, since each camera device must be able to stream the video to different destinations. It may be necessary for remote monitoring, for storage platforms to memorize multimedia data on the environment, or simply for local or remote systems to perform advanced processing like video tracking [2], activity recognition [3], intrusion detection, etc. This is shown in figure 2. In this case all the flows out from the camera to the different local and remote processes.

3 Sensor Manager

In the previous section we have described the different roles of the architecture that will operate over the camera devices and their associated control/video flows. In the pictures describing the architecture is introduced a *Sensor Manager* which is the responsible of attending the control flows, allowing the positioning of the PTZ cameras, and stream the video sequences to all locally-attached process and remote systems.

Therefore the *Sensor Manager* is not a trivial component of the architecture, as almost all the multi-agent systems suppose. It involves many disciplines like video acquisition, compression and transmission; control the protocols of the PTZ device and expose it to remote controlling processes, and so on.

So, in this section is described the *Sensor Manager* designed for this task. We have taken special care to the real-time restriction of the video surveillance systems, in which the video stream should be delivered with the minimum delay. The overall architecture of this controller is presented in figure 3.

This design let multiple access to a limited-access PTZ camera device, what usually only provides one serial communication port for control the orientation of the

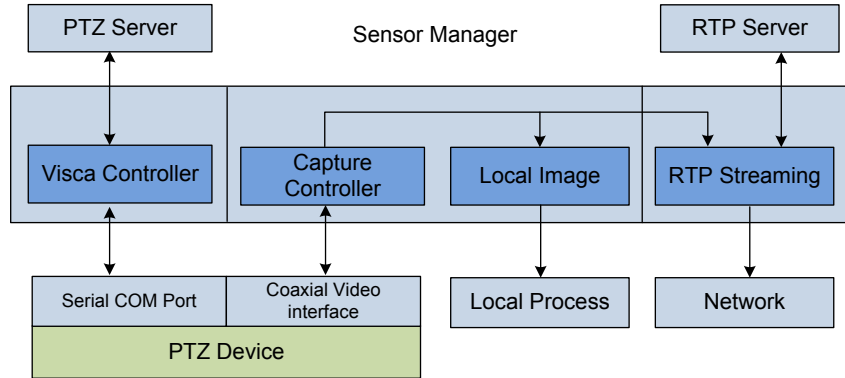


Fig. 3 Sensor manager for local/remote camera control

camera, and a coaxial video interface with the analog video signal. The sensor manager designed presents three different main functionalities which are described in more detail in the following subsections:

3.1 PTZ Controller

The PTZ controller designed allow us to provide an standard interface through the *PTZ Server* module to control homogeneously any underlying device. We have defined some high-level operating primitives like *goTo X Y*, *zoom amount*, etc, common in almost all PTZ devices. This commands will be interpreted by the proper controller (Visca Controller in the figure) and transmitted in the correct protocol to the device through the serial COM port.

High-level primitives are exposed by a non-connection oriented UDP Server, with a simple request-response protocol in the client-server computing mode. In our case we translate the high-level primitives incoming from the *PTZ Server* to the VISCA protocol [1], since all the cameras in our visual sensor network are compliant with this protocol. Moreover, any other device protocol may be used easily adapting a controller to our high-level interface.

3.2 Video Acquisition

Video acquisition from the device is a critical point of the sensor manager designed. In general, the image provided by present sensor devices comes in a analog format. This way is needed use digitizer cards to convert the signal to digital frames, in particular Matrox Morphis frame grabbers are used in our system. This introduces a handicap in the system, since the image provided by those cards is required by

both local and external processes, while these cards can be handled only by a single process.

To solve this issue we have defined two different strategies depending on the image destination. For local processes we have shared a region of non-paged memory which can be accessed by different processes of the local machine. So local processes can read the latest acquired frame directly from memory without any delay. Other local processes that needs to be notified when a new image is available could be attached directly (with a dynamic link library) to the sensor manager, as we do with a color-based object recognition implemented in the system.

For remote process we use other efficient way to provide frames with the minimum delay. In this case, Matrox Morphis boards allows JPEG2000 [7] compression in real-time, so we can obtain a compressed image version at the same time we acquire uncompressed frames (used for local processes or display). These JPEG2000 compressed frames are later transmitted (as described in following subsection) by a efficient and real-time streaming protocol [10].

3.3 Video Transmission

In order to transmit real-time video sequences to remote process like operators, agents, backup systems, etc, we have opted to implement a Real-time Transport Protocol (RTP) server based on JPEG2000 image sequences, described in the newly RFC5371 [6], inside the sensor manager.

The RTP defines a standardized packet format for delivering audio and video over IP networks. RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video conference applications and in general all those applications what needs a real-time communication.

In the design of the sensor manager architecture a RTP payload header extension has been implemented following the RFC5371. This standard defines a new RTP extension allowing transmit JPEG2000 frames (provided by Matrox Morphis boards) over RTP packets. Moreover in our implementation of the standard we have introduced a real-time motion compensation technique which is still patent pending (Spanish patent number P200900260) and still complaints with RFC5371. Hence, our RTP server is able to transmit real-time JPEG2000 frames to many clients, both in a unicast and multicast way.

4 Architecture evaluation

To evaluate the usability and performance of the architecture we have developed a functional prototype. It is divided in two different layers. The first layer is the sensor manager core, developed in different C++ modules (as described in the architecture) in order to meet real-time performance. The second one is the interface for config-

uring the core, as shown in figure 4. It provides both the RTP and PTZ servers, so any remote or local service may interact with the device concurrently. The presented Graphical User Interface (GUI) allows the user to select the capture device, camera channel, server ports, and some other parameters related with the JPEG2000 codec. It also enables a local PTZ controller for change the camera device orientation.

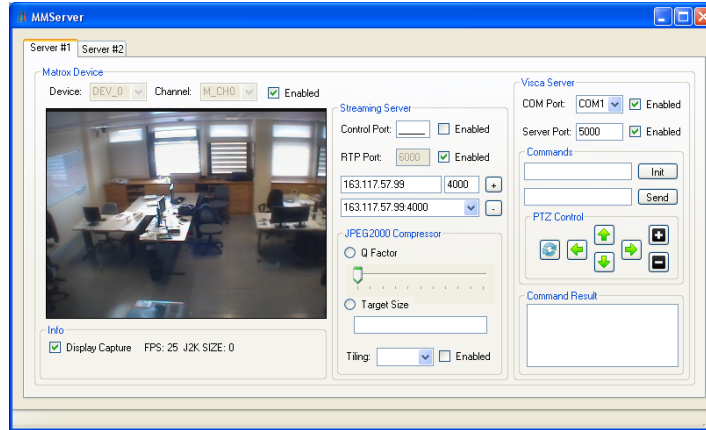


Fig. 4 Sensor Manager interface, providing controls over different aspects like RTP and PTZ servers

The sensor manager performance has been tested in two ways. In one hand we have measured the frames per second that the architecture is able to transmit both to local and remote processes. Each sensor manager controls one Matrox Morphis board (as the device only can be accessed by a single process at a time), and each board contains two hardware digitizers. It is possible to acquire video from 16 different sources, but in this case, it is necessary to share digitizers between all sources, hence the performance decreases. So we will use two sources for each board in order to find the real sensor manager performance, and not the limited by the underlying hardware.

Cameras	Local Process	Local/Remote Processes
One Camera	25 fps	25 fps
Two Cameras	25 fps	24-25 fps

Table 1 Transmission rate reached with the sensor manager architecture both for local and local/remote processes

Table 1 shows the different values obtained by the sensor manager in various operation modes, with one and two cameras transmitting video frames to local and remote processes. In both situations the frame-rate obtained is the same as the video sensor provides (25 FPS), so the architecture is working efficiently and in real-time

when transmitting the video frames. If the architecture not were able to process all the frames or send it across the network, the digitizer board will start do drop frames and the frame-rate will significantly be reduced.

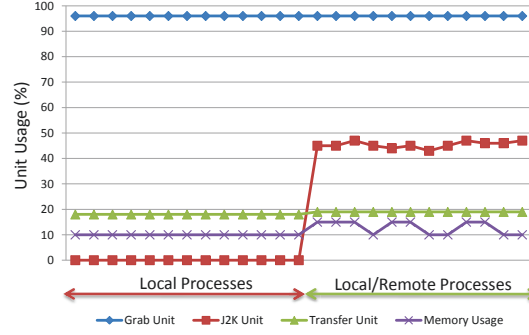


Fig. 5 Matrox Morphis unit usage under local and local/remote image transmission

We have also measured the Matrox Morphis board usage in order to confirm that the architecture is not adding extra lag in process frames. Figure 5 shows the usage of one digitizer when sharing frames with local processes and when sharing/transmitting frames to local/remote processes. In both cases the grab unit is always at 96%, meaning that all the video frames are being processed by the sensor manager. Notice also how the J2K unit start working only when is necessary transmitt frames to remote process.

In the other hand we have measured the total amount of time that the JPEG2000 streaming systems takes from when the image is captured in the server to when the image is displayed in the client. So it involves all the processes of acquisition, compression, transmission, decoding and displaying. The total latency obtained is about 180 milliseconds, which is adequate for real-time purposes like video surveillance, as discussed in [9].

Finally the PTZ server could not be evaluated since it process and execute all the high-level concurrent commands in sequence, so it depends on the commands executed, the camera movement speed, etc.

5 Conclusions

As outlined along this paper, the communication between local/remote processes with video sensor devices is so useful in the current growing visual sensor networks. It allows, among other things, to develop versatile architectures from cooperative multi-agent platforms, to sophisticated video surveillance systems. However the communication with a video device may be a unsatisfactory job since they used to have a limited-access and not always are accessible from remote places. An archi-

ture for controlling video devices from local and remote processes in a transparent way has been proposed in this paper to solve this issue. The test performed show that the architecture is able to efficiently share video frames to several destinations.

In future works we will include some high-level commands in the PTZ control, like 'follow the red bag'. In this case, the image processing like color tracking will be addressed locally.

6 Acknowledgments

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM CONTEXTS S2009/TIC-1485 and DPS2008-07029-C02-02

References

1. Sony EVI-D70/D70P Color Video Camera. *Technical Manual*.
2. J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 125–132. Citeseer, 2003.
3. R. Cilla, M. Patricio, A. Berlanga, and J. Molina. Fusion of Single View Soft k-NN Classifiers for Multicamera Human Action Recognition. *Hybrid Artificial Intelligence Systems*, pages 436–443, 2010.
4. G. Foresti, P. Mahonen, and C. Regazzoni. *Multimedia video-based surveillance systems: Requirements, issues, and solutions*. Springer Netherlands, 2000.
5. G. Foresti, C. Regazzoni, and P. Varshney. *Multisensor surveillance systems: The fusion perspective*. Kluwer Academic, 2003.
6. S. Futemma, E. Itakura, and A. Leung. RTP Payload Format for JPEG 2000 Video Streams. RFC 5371 (Informational), Oct. 2008.
7. ISO/IEC. 15444-1:2000 information technology jpeg2000 image coding system-part 1: core coding system. Technical report.
8. S. Kang, J. Paik, A. Koschan, B. Abidi, and M. Abidi. Real-time video tracking using PTZ cameras. In *Proc. of SPIE Vol.*, volume 5132, page 103. Citeseer.
9. G. Karlsson. Asynchronous transfer of video. *Communications Magazine, IEEE*, 34(8):118–126, 2002.
10. A. Luis and M. Patricio. Scalable Streaming of JPEG 2000 Live Video Using RTP over UDP. In *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*, pages 574–581. Springer.
11. J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. Jones. A multi-agent framework for visual surveillance. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 1104–1107. IEEE, 2002.
12. M. Patricio, J. Carbo, O. Perez, J. Garcia, and J. Molina. Multi-agent framework in visual sensor networks. *EURASIP Journal on Applied Signal Processing*, 2007(1):226, 2007.
13. S. Stillman, R. Tanawongsuwan, and I. Essa. A system for tracking and recognizing multiple people with multiple cameras. In *Proceedings of the Second International Conference on Audio-Vision-based Person Authentication*. Citeseer, 1999.
14. W. Wolf, B. Ozer, and T. Lv. Smart cameras as embedded systems. *Computer*, 35(9):48–53, 2002.